1. Write a program that asks the user to enter a list of at least five integers. Do the following:

(a) Print out the total number of items in the list.
(b) Print out the fourth item (index 3) in the list.
(c) Print out the last three items in the list.
(d) Print out all the items in the list except the first two.
(e) Print out the list in reverse order.
(f) Print out the largest and smallest values in the list.
(g) Print out the sum of all the values in the list.
(h) If the list contains a zero, print out the index of the first zero in the list, and otherwise print out a message saying there are no zeroes.
(i) Sort the list and print out the list after sorting.
(j) Delete the first item from the (now sorted) list and print out the new list.
(k) Change the second-to-last item in the list to 9876 and print out the new list.
(l) Append the value -500 to the end of the list and print out the new list.

In [1]:
```
def process_list(lst):
    # (a) Print out the total number of items in the list.
    print("Total number of items in the list:", len(lst))

    # (b) Print out the fourth item (index 3) in the list.
    print("Fourth item in the list:", lst[3])

    # (c) Print out the last three items in the list.
    print("Last three items in the list:", lst[-3:])

    # (d) Print out all the items in the list except the first two.
    print("All items in the list except the first two:", lst[2:])

    # (e) Print out the list in reverse order.
    print("List in reverse order:", lst[::-1])

    # (f) Print out the largest and smallest values in the list.
    print("Largest value in the list:", max(lst))
    print("Smallest value in the list:", min(lst))

    # (g) Print out the sum of all the values in the list.
    print("Sum of all values in the list:", sum(lst))

    # (h) If the list contains a zero, print out the index of the first zero in the list,
    #     and otherwise print out a message saying there are no zeroes.
    if 0 in lst:
        print("Index of the first zero in the list:", lst.index(0))
    else:
        print("There are no zeroes in the list.")

    # (i) Sort the list and print out the list after sorting.
    sorted_lst = sorted(lst)
    print("List after sorting:", sorted_lst)

    # (j) Delete the first item from the (now sorted) list and print out the new list.
    del sorted_lst[0]
    print("List after deleting the first item:", sorted_lst)

    # (k) Change the second-to-last item in the list to 9876 and print out the new list.
    sorted_lst[-2] = 9876
    print("List after changing the second-to-last item:", sorted_lst)

    # (l) Append the value -500 to the end of the list and print out the new list.
    sorted_lst.append(-500)
    print("List after appending -500:", sorted_lst)
```

Write a program that asks the user to enter a list of numbers. Then print out the smallest thing in the list and the first index at which it appears in the list.

In [6]:
```
numbers = input("Enter a list of numbers, separated by spaces: ").split()
numbers = [float(num) for num in numbers]

smallest = min(numbers)
index = numbers.index(smallest)

print("Smallest number:", smallest)
print("First index of smallest number:", index)
```

```
Enter a list of numbers, separated by spaces: 5 4 89 74 9
Smallest number: 4.0
First index of smallest number: 1
```

Write a program that asks the user to enter a string of lowercase letters and creates a list containing counts of how many times each letter appears in the string. The first index is how many a's are in the string, the second is how many b's, etc.

In [8]:
```
def count_letters(string):

    counts = [0] * 26


    for char in string:

        if char.islower():
            # Increment the count for the corresponding letter by 1
            counts[ord(char) - ord('a')] += 1

    return counts



user_input = input("Enter a string of lowercase letters: ")


letter_counts = count_letters(user_input)


for i in range(26):
    letter = chr(ord('a') + i)
    count = letter_counts[i]
    print(f"Number of {letter}'s: {count}")
```

```
Enter a string of lowercase letters: hello world
Number of a's: 0
Number of b's: 0
Number of c's: 0
Number of d's: 1
Number of e's: 1
Number of f's: 0
Number of g's: 0
Number of h's: 1
Number of i's: 0
Number of j's: 0
Number of k's: 0
Number of l's: 3
Number of m's: 0
Number of n's: 0
Number of o's: 2
Number of p's: 0
Number of q's: 0
Number of r's: 1
Number of s's: 0
Number of t's: 0
Number of u's: 0
Number of v's: 0
Number of w's: 1
Number of x's: 0
Number of y's: 0
Number of z's: 0
```

Create a dictionary whose keys are the strings 'abc', 'def', 'ghi', 'jkl', and 'mno' and whose corresponding values are 7, 11, 13, 17, and 19. Then write dictionary code that does the following:

(a) Print the value in the dictionary associated with the key 'def'.
(b) Use the keys() method to print out all the keys.
(c) Loop over the dictionary and print out all the keys and their associated values.
(d) Use an if statement to check if the dictionary contains the key 'pqr' and print out an appropriate

message indicating whether it does or doesn't.
(e) Change the value associated with the key 'abc' to 23 and then print out all the values in the dictionary using the values() method.

In [9]:
```
# Create the dictionary
my_dict = {'abc': 7, 'def': 11, 'ghi': 13, 'jkl': 17, 'mno': 19}

# (a) Print the value associated with the key 'def'
print(my_dict['def'])

# (b) Print all the keys using keys() method
print(my_dict.keys())

# (c) Loop over the dictionary and print keys and their associated values
for key, value in my_dict.items():
    print(f"Key: {key}, Value: {value}")

# (d) Check if the dictionary contains the key 'pqr' and print appropriate message
if 'pqr' in my_dict:
    print("The dictionary contains the key 'pqr'")
else:
    print("The dictionary does not contain the key 'pqr'")

# (e) Change the value associated with the key 'abc' to 23 and print all values using values() method
my_dict['abc'] = 23
print(my_dict.values())
```

```
11
dict_keys(['abc', 'def', 'ghi', 'jkl', 'mno'])
Key: abc, Value: 7
Key: def, Value: 11
Key: ghi, Value: 13
Key: jkl, Value: 17
Key: mno, Value: 19
The dictionary does not contain the key 'pqr'
dict_values([23, 11, 13, 17, 19])
```

In [ ]: